IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

| | |
|---|---|
| Applicant | : McGarvey et al. |
| Serial No. | : 09/921,536 |
| Filed | : August 03, 2001 |
| Title | : Methods, Systems and Computer Program Products For Secure Delegation Using Public Key Authorization |
| Attorney Docket | : 5577-236 (RSW920000185US1-IBM 019 PA) |
| Examiner | : M. Henning |
| Art Unit | : 2131 |
| Confirm | : 6803 |

Mail Stop Appeal Brief Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

## APPELLANT'S BRIEF ON APPEAL UNDER 37 C.F.R. §41.37

Sir:

This Appeal Brief is filed pursuant to the "Notice of Appeal to the Board of Patent Appeals and Interferences", filed November 17, 2006 and the Notice of Panel Decision from Pre-Appeal Brief Review, mailed December 20, 2006.

### Real Party In Interest

The Real Party in Interest in the present Appeal is International Business Machines Corporation Armonk, New York, the assignee, as evidenced by the assignment set forth at Reel/Frame 012073/0180.

### Related Appeals and Interferences

The appellant is aware of no appeals or interferences that would be affected by the present appeal.

### Status of the Claims

Claims 1-32 stand finally rejected by the Examiner as noted in the Final Office Action mailed August 18, 2006 and corresponding Advisory Action mailed November 08, 2006. The rejection of claims 1-32 is appealed. The claims at issue are attached hereto as Appendix A.

## Status of Amendments

Appellants filed a Request for Reconsideration on October 24, 2006 in response to an Office action made Final, which was mailed on August 18, 2006. The Request for Reconsideration included a Declaration under 37 C.F.R. §1.131 that swore behind the primary reference, U.S. Pat. App. Pub. No. 2003/0018913 to Brezak et al. No claims were amended or canceled in the Response. The Examiner did not enter the Request For Reconsideration or the associated declaration as evidenced by the Advisory Action, which was mailed on November 08, 2006.

## Summary of Claimed Subject Matter

The claimed invention is directed to the authentication of a client when delegation is utilized to access a server. Certain networked computer systems are deployed using a "tiered" model where the first tier, e.g., corresponding to a client or principal, communicates with a second (middle) tier server, e.g., a web server. The middle tier server may be required to communicate with a third tier server, e.g., a back-end server such as a database server, etc., on behalf of the client. Under certain circumstances, it may be necessary to propagate security credentials of the client (referred to as "delegation" or "impersonation") through one or more of the tiers for purposes of authentication[1]. However, conventional authorization techniques are problematic when delegation is necessary, such as where there are multiple servers for which the client may need authentication[2].

As such, the claimed invention provides "a common nonce" that is associated with each of a plurality of servers, e.g., third tier and beyond, and which is signed by the client. Because each of the plurality of servers contributes to the "common nonce" and that common nonce is signed by the client, each of the plurality of servers can authenticate the client. As such, the

[1] Page 1, line 19 – Page 2, line 3.
[2] A conventional approach for authentication of security credentials uses a standard Public Key Infrastructure (PKI) authentication process, which uses a ***nonce***, e.g., a random number generated by or on behalf of a server. The nonce is given to the client, which signs the nonce with its digital signature, and returns the signed nonce to the server. The signed nonce is decrypted using a corresponding public key of the client. If the decrypted nonce matches the originally sent nonce, then the client is authenticated. (See Page 2, lines 4-25)

middle tier server can easily delegate (impersonate the client) to any one or more of the plurality of servers.

***Independent claim 1*** is directed to a method for a middle-tier server to impersonate a client to a plurality of servers (See for example, page 2, lines 28-30), the method comprising:

obtaining a common nonce associated with each of the plurality of servers from an entity other than the client or the plurality of servers; (e.g., Fig. 1A exchange between the middle tier server 14 and plurality of servers 20, 22, 24 and corresponding Page 8, lines 21-26; Fig. 1B exchange between the third party 18 and plurality of servers 20, 22, 24 and corresponding Page 9, lines 8-14; Fig. 5 and corresponding Page 12, line 22-Page 13, line 24)

providing the common nonce to the client; receiving the common nonce signed by the client at the middle-tier server; (e.g., Fig. 1A and corresponding Page 8, lines 26-30; Fig. 1B and corresponding Page 9, lines 14-19; Fig. 4 and corresponding Page 12, lines 4-15)

and

providing the signed common nonce to the plurality of servers as a signature for transactions so as to authenticate the client to the plurality of servers (e.g., Box 415 of Fig. 4; Box 615 of Fig. 6; Fig. 7 and corresponding Page 9, line 12-Page 13, line 13; Fig. 9 and corresponding Page 15, line 26-Page 16, line 6).

**Independent claim 26** is directed to a system for a middle-tier server to impersonate a client to a plurality of servers, comprising:

means for obtaining a common nonce associated with each of the plurality of servers from an entity other than the client or the plurality of servers; (e.g., any one or more of: the nonce processing module 360 and corresponding pre-nonce values 362 and/or nonce values 364 in Fig. 3 and corresponding Page 11, lines 11-18; the I/O data ports 246 coupled to the processor 238 in Fig. 2 and corresponding Page 10, lines 4-14; I/O device drivers 358 in Fig. 3 and corresponding description of exemplary communications protocols at Page 10, line 24-Page 11, line 10; Fig. 5 and corresponding Page 12, line 22-Page 13, line 24)

means for providing the common nonce to the client; means for receiving the common nonce signed by the client at the middle-tier server; (e.g., any one or more of: the nonce processing module 360 in Fig. 3 and corresponding Page 11, lines 11-18; the I/O data ports 246

coupled to the processor 238 in Fig. 2 and corresponding Page 10, lines 4-14; I/O device drivers 358 in Fig. 3 and corresponding description of exemplary communications protocols at Page 10, line 24-Page 11, line 10; the communication of the common nonce to the client 10 and return of the signed common nonce from the client 10 to the middle tier server 14 in Figs. 1A, 1B and corresponding Page 8, lines 21-26 and Page 9, lines 8-14; Fig. 4 and corresponding Page 12, lines 4-15) and

means for providing the signed common nonce to the plurality of servers as a signature for transactions so as to authenticate the client to the plurality of servers. (e.g., any one or more of: the nonce processing module 360 in Fig. 3 and corresponding Page 11, lines 11-18; the I/O data ports 246 coupled to the processor 238 in Fig. 2 and corresponding Page 10, lines 4-14; I/O device drivers 358 in Fig. 3 and corresponding Page 10, line 24-Page 11, line 11; Box 415 of Fig. 4; Box 615 of Fig. 6.

**Independent claim 27** is directed to a computer program product for a middle-tier server to impersonate a client to a plurality of servers, comprising:

a computer readable media having computer readable program code embodied therein, the computer readable program code comprising: (See for example, Page 6, lines 18-28)

computer readable program code that obtains a common nonce associated with each of the plurality of servers from an entity other than the client or the plurality of servers; (e.g., Fig. 1A exchange between the middle tier server 14 and plurality of servers 20, 22, 24 and corresponding Page 8, lines 21-26; Fig. 1B exchange between the third party 18 and plurality of servers 20, 22, 24 and corresponding Page 9, lines 8-14; Fig. 5 and corresponding Page 12, line 22-Page 13, line 24)

computer readable program code that provides the common nonce to the client; computer readable program code that receives the common nonce signed by the client at the middle-tier server; (e.g., Fig. 1A and corresponding Page 8, lines 26-30; Fig. 1B and corresponding Page 9, lines 14-19; Fig. 4 and corresponding Page 12, lines 4-15) and

computer readable program code that provides the signed common nonce to the plurality of servers as a signature for transactions so as to authenticate the client to the plurality of servers (e.g., Box 415 of Fig. 4; Box 615 of Fig. 6; Fig. 7 and corresponding Page 9, line 12-Page 13, line 13; Fig. 9 and corresponding Page 15, line 26-Page 16, line 6).

**Independent claim 28** is directed to a method of authenticating a client, comprising:

receiving at a server of a plurality of servers, a common nonce that is provided to each of the plurality of servers from an entity other than the client or the plurality of servers, the common nonce being associated with each of the plurality of servers and signed by the client; (e.g., Page 8, lines 27-30; Page 9, lines 15-19; Box 415 of Fig. 4; Box 615 of Fig. 6) and

authenticating the client based on the received signed common nonce (e.g., Fig. 7 and corresponding Page 9, line 12-Page 13, line 13; Fig. 9 and corresponding Page 15, line 26-Page 16, line 6).

**Independent claim 31** is directed to a system for authenticating a client, comprising:

means for receiving at a server of a plurality of servers, a common nonce that is provided to each of the plurality of servers from an entity other than the client or the plurality of servers, the common nonce being associated with each of the plurality of servers and signed by the client; ( e.g., one or more of: the nonce processing module 360 in Fig. 3 and corresponding Page 11, lines 11-18; the I/O data ports 246 coupled to the processor 238 in Fig. 2 and corresponding Page 10, lines 4-14; I/O device drivers 358 in Fig. 3 and corresponding Page 10, line 24-Page 11, line 10; Fig. 1A and corresponding Page 8, lines 26-30; Fig. 1B and corresponding Page 9, lines 14-19; Fig. 4 and corresponding Page 12, lines 4-15) and

means for authenticating the client based on the received signed common nonce (e.g., one or more of: the nonce processing module 360 in Fig. 3 and corresponding Page 11, lines 11-18; the I/O data ports 246 coupled to the processor 238 in Fig. 2 and corresponding Page 10, lines 4-14; I/O device drivers 358 in Fig. 3; Fig. 7 and corresponding Page 9, line 12-Page 13, line 13; Fig. 9 and corresponding Page 15, line 26-Page 16, line 6).

**Independent claim 32** is directed to a computer program product for authenticating a client, comprising:

a computer readable media having computer readable program code embodied therein, the computer readable program code comprising: (e.g., Page 6, lines 18-28)

computer readable program code which receives at a server of a plurality of servers, a common nonce that is provided to each of the plurality of servers from an entity other than the

client or the plurality of servers, the common nonce being associated with each of the plurality of servers and signed by the client; ; (e.g., Fig. 1A and corresponding Page 8, lines 26-30; Fig. 1B and corresponding Page 9, lines 14-19; Fig. 4 and corresponding Page 12, lines 4-15) and

computer readable program code which authenticates the client based on the received signed common nonce (e.g., Fig. 7 and corresponding Page 9, line 12-Page 13, line 13; Fig. 9 and corresponding Page 15, line 26-Page 16, line 6).

## Grounds of Rejection To Be Reviewed On Appeal

1. Whether Claims 1, 23-29 and 31-32 are unpatentable under 35 U.S.C. §103(a) as being obvious over U.S. Pat. App. Pub. No. 2003/0018913 to Brezak et al. (hereinafter '*Brezak*') in view of U.S. Pat. No. 5,535,276 to Ganesan (hereinafter '*Ganesan*').

2. Whether Claims 2, 3, 5, 7-11, 14, 15 and 30 are unpatentable under 35 U.S.C. §103 as being obvious over *Brezak* in view of *Ganesan* and further in view of U.S. Pat. No. 6,829,356 to Ford (hereinafter '*Ford*').

3. Whether Claims 4, 6, 12, 13 and 20 are unpatentable under 35 U.S.C. §103 a being obvious over *Brezak* in view of *Ganesan*, *Ford* and further in view of "Applied Cryptography" by Schneier (hereinafter '*Schneier*').

4. Whether Claims 16-19 and 21-22 are unpatentable under 35 U.S.C. §103 a being obvious over *Brezak* in view of *Ganesan*, *Ford* and further in view of "Handbook of Applied Cryptography" by Menezes et al. (hereinafter '*Menezes*').

## Arguments

### Introduction to 35 U.S.C. §103(a) Analysis

According to the M.P.E.P. §706.02(j), to establish a *prima facie* case of obviousness, the prior art reference must teach or suggest all the claim limitations. It is the appellant's position that a *prima facie* case of obviousness has not been established for any of the claims, because the cited references, even when combined, fail to teach or suggest all of the limitations of any pending claim, as set out more fully herein.

**1. Claims 1, 23-29 and 31-32 are Patentable Over *Brezak* in view of *Ganesan***

Of the Claims rejected under the first ground of rejection, Claims 1, 26, 27, 28, 31 and 32 are in independent form. Each claim recites obtaining or receiving a common nonce associated with a plurality of servers. Nowhere in the four corners of either *Brezak* or *Ganesan* is a nonce even mentioned, let alone a common nonce as claimed.

**1-A. Independent Claims 1 is patentable over *Brezak* and *Ganesan***

Claim 1 recites in pertinent part:

> obtaining a common nonce associated with each of the plurality of servers from an entity other than the client or the plurality of servers;
> providing the common nonce to the client;
> receiving the common nonce signed by the client at the middle-tier server;
> providing the signed common nonce to the plurality of servers as a signature for transactions so as to authenticate the client to the plurality of servers.

The Office Action asserts that *Brezak* discloses all of the recitations of Claim 1 with the exception of "…receiving the common nonce signed by the client at the middle-tier server". The Office action further asserts that *Ganesan* teaches that to prevent dictionary attacks, a ticket can be encrypted by a ticket granting system with a key that is shared between the server and the ticketing system[3]. The Examiner thus concludes that it would be obvious to combine the encryption of *Ganesan* with the system of *Brezak* to prevent dictionary attacks[4]. In support of this position, the Examiner attempts to read the "privilege attribute certificate" (PAC) disclosed by *Brezak* onto the claimed "…common nonce associated with each of the plurality of servers from an entity other than the client or the plurality of servers"[5].

Contrary to the Examiner's conclusion, it is the appellant's position that, when reading claim 1 *as a whole*, *Brezak* combined with *Ganesan* fail to teach or suggest providing a common nonce (signed by a client) to a plurality of servers as a signature for transactions so as to authenticate the client to the plurality of servers. Moreover, it is the appellant's position that,

---

[3] See Office Action mailed 08/18/2006, starting at Page 3, line 17 to Page 4, line 5. .
[4] See Office Action mailed 08/18/2006, Page 4.
[5] See Office Action mailed 08/18/2006, Pages 2-3.

despite the Examiner's assertions, *Brezak* can not reasonably be read so as to teach or suggest a common nonce as claimed.

Combination Of *Brezak* and *Ganesan* Does Not Teach Every Claim Element

Even assuming *arguendo* that a PAC as taught by *Brezak* (explained in greater detail below) can somehow be considered a nonce, which the appellant rigorously asserts it cannot, the references still fail to teach of suggest the claimed invention because even if the PAC is somehow signed by a client as the Examiner suggests via the combination of *Ganesan* to *Brezak*, that PAC is never provided to the plurality of servers as a signature for transactions so as to authenticate the client to the plurality of servers, as claimed.

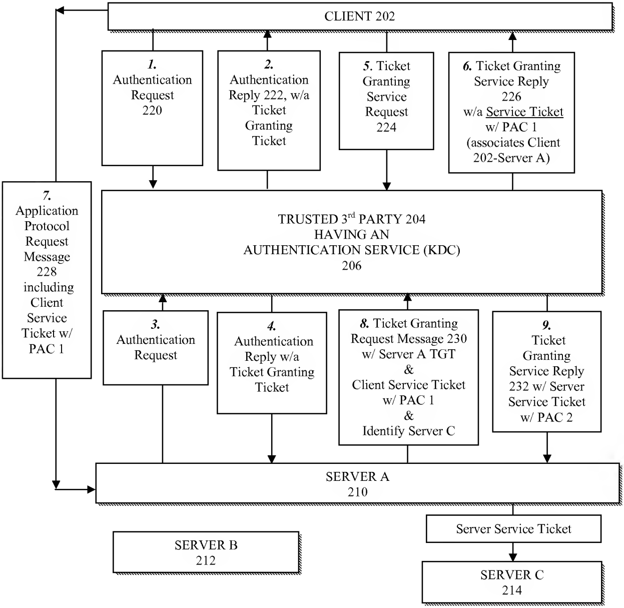
In order to see this, it is important to understand the transaction method taught by *Brezak*. *Brezak* discloses systems for controlling user authentication within a Kerberos environment[6]. A communication flow is illustrated in the Figure below so that the various communications and sequence of communications disclosed by *Brezak* can be readily understood. As disclosed in *Brezak*, in order to participate in a Kerberos environment, both a client 202 and a front-end Server A must identify themselves as known and registered participants to the trusted Kerberos server. To do this, the client 202 logs onto the network and sends an "authentication request" message 220 to an authentication service 206 of a trusted third party 204 (see Message 1). The authentication service 206 replies to the authentication request with an authentication reply message 222, which includes a "Ticket Granting Ticket" (See Message 2). In an analogous manner, Server A also sends an authentication request and obtains a Ticket Granting Ticket from the authentication service (See Messages 3 and 4)[7].

At this point, both the client 202 and Server A have identified themselves to the trusted third party and are confirmed as "belonging" to the Kerberos environment. However, there have been no communications between the client and Server A.

---

[6] Kerberos is a network control protocol. See for example, *Brezak*, paragraph 3.
[7] See *Brezak*, paragraph 42.

When the client 202 wants to communicate with Server A, the client sends a Ticket Granting Service Request Message 224 to the third party authentication service 206 (see Message 5) indicating its desire to communicate with Server A. The third party authentication service 206 replies to the Request Message 224 with a Ticket Granting Service Reply 226 (see Message 6). The Ticket Granting Service Reply includes a *client service ticket* that associates the client with Server A and may also include a PAC[8].




---

[8] See Brezak, paragraphs 43, 49.

The client then forwards its client service ticket to Server A in an Application Protocol Request Message 228 (see Message 7)[9]. Of significance, in a conventional Kerberos transaction, the client would have sent its Ticket Granting Ticket (which it received in its Authentication reply from the third party Authentication service – Message 2). Having possession of the client's ticket granting ticket would allow unconstrained delegation (unconstrained approval of requests for service tickets to delegate) for the life of that ticket granting ticket. However, the system in *Brezak* intentionally constrains delegation by intentionally *not* distributing the client's ticket granting ticket[10]. Rather, as will be seen, in *Brezak*, a server is constrained and must request and be approved for each new service ticket to delegate to a new server on behalf of the client[11].

In the disclosed example in *Brezak* as described with reference to Fig. 2, it is assumed that Server A needs to access Server C on behalf of the client 202[12]. Thus, Server A sends a Ticket Granting Service Request Message 230 to the third party authentication service 206 (see message 8), which includes Server A's previously obtained Ticket Granting Ticket, the *client's service ticket* and the identity of the target server (Server C in this example)[13].

The third party authentication service 206 checks to see if the client 202 has authorized delegation in response to receiving the ticket granting service message from Server A. The authentication service 206 replies to Server A with a Ticket Granting Service Reply Message 232 that includes *a new* service ticket for Server A (see message 9) *if* the authentication service 206 determines that it is okay to delegate. The reply message *may also* provide client account data in the form of a privilege attribute certificate (PAC).

There are two possible sources of the PAC information in *Server A's service ticket*. The PAC may be derived by the third party authentication service, e.g., from an authentication

[9] See *Brezak*, paragraph 43.
[10] See *Brezak*, paragraphs 6; 35.
[11] See *Brezak*, paragraph 36
[12] See *Brezak*, paragraph 44.
[13] See *Brezak*, paragraph 44.

database 208, or the authentication service may simply copy relevant PAC data from the client's service ticket if that ticket included the PAC information[14].

Thus, the PAC that Service A receives as part of its service ticket (see message 9 of the Figure herein, which is necessary to transact with Server C according to Kerberos protocol) may be originated by the third party authentication service 206. However, under this arrangement, the PAC *cannot teach or suggest* "... providing the common nonce to the client...receiving the common nonce signed by the client at the middle-tier server" as claimed because the client never sees nor has the opportunity to sign the PAC included with the service ticket sent to Server A.

On the other hand, the third party authentication service 206 may copy relevant PAC data from the client's service ticket if that ticket included the PAC information. Assume *arguendo* that the client digitally signs the Application Protocol Request Message 228 and/or the included Client Service Ticket w a PAC (message 7), such as using a cryptographic method[15] as taught in *Ganesan* as the Examiner argues. This also fails to teach or suggest the claimed invention. For example, claim 1 recites "... providing the signed common nonce to the plurality of servers as a signature for transactions so as to authenticate the client to the plurality of servers".

As the flow provided in the Figure herein illustrates, the now "signed" Application Protocol Request Message (message 7) as the Examiner would have it, has to be parsed to extract the client's service ticket and corresponding PAC information. Thus, either Server A or the third party authentication service 206 *must decrypt* the message using, for example, the decrypting scheme of the split private key asymmetric cryptography as taught in *Ganesan*. The third party authentication service 206 must then build the service ticket for Server A using information from the (now decrypted) PAC of the client's Application Protocol Request Message 228. (In this regard, it is noted that the additional "layers" of encryption may already be in place according to the Kerberos protocol, in addition to that suggested by the Examiner's combination of *Brezak* and *Ganesan*).

---

[14] See *Brezak*, , paragraphs 45-49.

[15] *Ganesan* is directed to split private key asymmetric cryptography, where a message, including a ticket to access a server 50, is encrypted/signed and then verified to authenticate a client 10 to a server 50.

Thus, even under this configuration, the client's service ticket and PAC *are not* transferred between the client, Server A and Server C, regardless of whether the client signs the Application Protocol Request message or not. Rather, any PAC that the client has access to is passed through Server A to the third party authentication service 206. The third party authentication service 206 reads the clients service ticket information as well as other information provided by Server A to determine whether or not delegation is even *allowed* (see message 8). Assuming it is allowed, a *new* service ticket and PAC are generated by the third party authentication service 206 (see message 9), which is returned to Server A for subsequent delegation to Server C. This is the very essence of the modification to the Kerberos environment taught by *Brezak*, and which is necessary to constrain delegation.

Moreover, as illustrated in the Figure provided herein, it is the responsibility of the trusted third party authentication service to authenticate the client to the system. Indeed, this is the initial step (See messages 1-4) that is performed in a Kerberos environment and occurs before any transactions between servers. As such, any back-end servers, e.g., Server C would not have to independently authenticate the client, as authentication is assumed to have been completed by virtue of the granting of a service ticket.

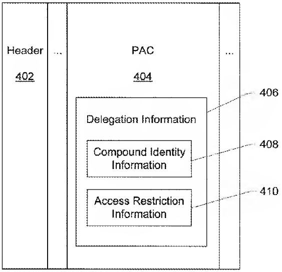
The appellant respectfully asserts that regardless of how the information is obtained, the PAC and service ticket received by Server A for purposes of delegating to a subsequent server in *Brezak* is *always* created by the third party authentication service 206 and is given directly to Server A. Server A then uses its new service ticket to request access to Server C using the Kerberos protocols as described more fully herein[16]. Therefore, even if *Ganesan* is somehow combined with *Brezak* as the Examiner suggests, the argued combination still fails to teach each claim limitation.

---

[16] See *Brezak*, paragraph 55.

<u>*A PAC is not a Nonce as Claimed*</u>

A privilege attribute certificate (PAC) 404 as taught by *Brezak*[17], reproduced below, is included as part of a Keberos message 400 and includes delegation information 406 such as a "compound identity information" field 408 and an "access restriction information" field 410. The compound identity information field stores a history of servers that request a *service ticket* on behalf of the client. Thus, a history can be reconstructed over multiple delegation processes[18]. The access restriction information is utilized in conjunction with an "access control mechanism" (which is not further described in *Brezak*) to selectively allow access to certain servers/services that the associated client has either directly, or indirectly designated through the first server[19].




As an example, if Server A is the first (intermediate) server, and the client has either directly or indirectly designated Server A to delegate to back-end Server C, such as via a suitable entry (or lack of entry depending upon the logic) in the access restriction information, then Server A can be granted permission to transact with Server C on behalf of the client. However, Server B cannot delegate to Server C on behalf of the client.

As *claimed*, a common nonce associated with a plurality of servers is provided to a client. The client signs the common nonce and returns it to a middle tier server, and the signed common

[17] See Fig. 4 of *Brezak,* included herein.
[18] See *Brezak*, paragraph 51.
[19] See *Brezak*, paragraphs 50-52.

nonce is provided to a plurality of servers as a signature for transactions so as to authenticate the client to the plurality of servers. Thus, for example, the claimed signed common nonce is provided to *authenticate* a client to a plurality of (back-end) servers.

In a Kerberos system such as taught in *Brezak*, *authentication* is performed by the exchange and verification of the (typically encrypted) Service ticket by the third party authentication service 206 (not the back-end servers themselves). The PAC on the other hand is a set of data fields maintained by the third party authentication service 206 for tracking the history and access restrictions of requests for service tickets on behalf of a client, and thus is used to restrict *delegation*. Moreover, because the PAC tracks historical information, each time the PAC is copied by the third party authentication service 206, it is likely to have new, additional and/or different information. As such, the content of the PAC issued by the third party authentication to Server A (see message 9) is likely to have different information than a PAC provided to Server A in the client's Application Protocol Request Message 228 (see message 7). Thus a PAC does not teach or suggest a *nonce* at all.

Accordingly, when considering claim 1 *as a whole*, the prior art references fail to teach or suggest all of the claimed limitations. For the reasons set out above, the Board is respectfully requested to reverse the Examiner's final rejection of claim 1 and corresponding dependent claims 23-25.

**1-B. Independent Claim 26 is patentable over *Brezak* and *Ganesan***

Claim 26 includes claim elements that are similar to those recited in claim 1 such that the arguments presented for claim 1 apply to claim 26 by analogy. For example, as noted in greater detail above, neither *Brezak* nor *Ganesan* teach or suggest a common nonce associated with a plurality of servers. As such, the references, even when combined still fail to teach or suggest "…means for obtaining a common nonce associated with each of the plurality of servers from an entity other than the client or the plurality of servers" as claimed.

Still further, as noted in greater detail above, neither *Brezak* nor *Ganesan* teach or suggest a providing a common nonce (signed by a client) to a plurality of servers. As such, the

references, even when combined, fail to teach or suggest "…means for providing the common nonce to the client…means for receiving the common nonce signed by the client at the middle-tier server …means for providing the signed common nonce to the plurality of servers as a signature for transactions so as to authenticate the client to the plurality of servers" as claimed.

Accordingly, when considering claim 26 *as a whole*, the prior art references fail to teach or suggest all of the claimed limitations. For the reasons set out above, the Board is respectfully requested to reverse the Examiner's final rejection of claim 26.

**1-C. Independent Claim 27 is patentable over *Brezak* and *Ganesan***

Claim 27 includes claim elements that are similar to those recited in claim 1 such that the arguments presented for claim 1 apply to claim 27 by analogy. For example, as noted in greater detail above, neither *Brezak* nor *Ganesan* teach or suggest a common nonce associated with a plurality of servers. As such, the references, even when combined still fail to teach or suggest "…computer readable program code that obtains a common nonce associated with each of the plurality of servers from an entity other than the client or the plurality of servers" as claimed.

Still further, as noted in greater detail above, neither *Brezak* nor *Ganesan* teach or suggest a providing a common nonce (signed by a client) to a plurality of servers. As such, the references, even when combined, fail to teach or suggest "…computer readable program code that provides the common nonce to the client … computer readable program code that receives the common nonce signed by the client at the middle-tier server … computer readable program code that provides the signed common nonce to the plurality of servers as a signature for transactions so as to authenticate the client to the plurality of servers." as claimed.

Accordingly, when considering claim 27 *as a whole*, the prior art references fail to teach or suggest all of the claimed limitations. For the reasons set out above, the Board is respectfully requested to reverse the Examiner's final rejection of claim 27.

**1-D. Independent Claim 28 is patentable over *Brezak* and *Ganesan***

Claim 28 recites in pertinent part:

receiving at a server of a plurality of servers, a common nonce that is provided to each of the plurality of servers from an entity other than the client or the plurality of servers, the common nonce being associated with each of the plurality of servers and signed by the client; and

authenticating the client based on the received signed common nonce.

As noted in greater detail above, in the system as taught by *Brezak* and/or *Ganesan* (they are both Kerberos systems), there is no teaching or suggestion of a nonce. Moreover, even if a nonce is disclosed, there is no teaching or suggestion of receiving at a server of a plurality of servers, a common nonce that is provided to each of the plurality of servers from an entity other than the client or the plurality of servers, the common nonce being associated with each of the plurality of servers and signed by the client.

As noted in the discussion of *Brezak* above, in order for Server A to communicate on behalf of a client 202 with Server C (e.g., Server C corresponding to the claimed "…server of a plurality of servers"), Server A must obtain its own Service ticket from the third party authentication service 206 (see message 9). However, the information provided in that service ticket is *created by* the third party authentication service 206. Thus, the third party authentication service 206 is used as a single trusted source for both authentication and authorization.

Moreover, there is no teaching or suggestion that a common nonce is provided to each of the plurality of servers. Rather, in *Brezak* and in *Ganesan* (both references disclose Kerberos systems), each time a middle tier server, e.g., Server A in *Brezak*, wants to communicate with any one of the back-end servers, e.g,. Server B, Server C, Server D, *a new service ticket* is required. This requires Service A to request a service ticket from the third party authentication service 206 for each new server that it wants to transact with on behalf of the client[20]. As noted more fully herein, this further likely results in a PAC having different information each time a service ticket is generated by the third party authorization service 206.

[20] See for example, *Brezak*, paragraph 45.

As such, even assuming *arguendo*, that a PAC received by Server A (in message 7) could be considered a signed nonce as the Examiner argues in his combination of *Brezak* and *Ganesan*, it is not provided from Server A to a single additional back-end server, let alone to each of the plurality of servers as claimed, because Server A is required to get a service ticket for each transaction with a different Server. As the PAC contains historical information as well as delegation information, the PAC provided in each service ticket will likely be different.

Moreover, the third party authentication service 206 may refuse granting a service ticket for a particular server, such as where the associated PAC information indicates that the client did not authorize Server A to delegate to a particular back-end server[21].

Accordingly, when considering claim 28 *as a whole*, the prior art references fail to teach or suggest all of the claimed limitations. For the reasons set out above, the Board is respectfully requested to reverse the Examiner's final rejection of claim 28 and corresponding dependent claims 29 and 30.

**1-E. Independent Claim 31 is patentable over *Brezak* and *Ganesan***

Claim 31 includes claim elements that are similar to those recited in claim 28 such that the arguments presented for claim 28 apply to claim 31 by analogy. For example, as noted in greater detail above, neither *Brezak* nor *Ganesan* teach or suggest a common nonce. As such, the references, even when combined still fail to teach or suggest "…means for receiving at a server of a plurality of servers, a common nonce that is provided to each of the plurality of servers from an entity other than the client or the plurality of servers, the common nonce being associated with each of the plurality of servers and signed by the client" as claimed.

Still further, as noted in greater detail above, neither *Brezak* nor *Ganesan* teach or suggest a common nonce that is provided to each of the plurality of servers. As such, the references, even when combined, fail to teach or suggest "…means for receiving at a server of a plurality of servers, a common nonce that is provided to each of the plurality of servers" as claimed.

---

[21] See for example, *Brezak*, paragraph 46.

Still further, as noted in greater detail above, neither *Brezak* nor *Ganesan* teach or suggest that a server authenticates the client. Rather, as described in greater detail herein, authentication is performed by the third party authentication service 206 before granting the intermediate server (Server A) permission to delegate on behalf of the client. As such, the references, even when combined still fail to teach or suggest "…means for authenticating the client based on the received signed common nonce" as claimed.

Accordingly, when considering claim 31 *as a whole*, the prior art references fail to teach or suggest all of the claimed limitations. For the reasons set out above, the Board is respectfully requested to reverse the Examiner's final rejection of claim 31.

**1-F. Independent Claim 32 is patentable over *Brezak* and *Ganesan***

Claim 32 includes claim elements that are similar to those recited in claim 28 such that the arguments presented for claim 28 apply to claim 32 by analogy. For example, as noted in greater detail above, neither *Brezak* nor *Ganesan* teach or suggest a common nonce. As such, the references, even when combined still fail to teach or suggest "…computer readable program code which receives at a server of a plurality of servers, a common nonce that is provided to each of the plurality of servers from an entity other than the client or the plurality of servers, the common nonce being associated with each of the plurality of servers and signed by the client" as claimed.

Still further, as noted in greater detail above, neither *Brezak* nor *Ganesan* teach or suggest a common nonce that is provided to each of the plurality of servers. As such, the references, even when combined, fail to teach or suggest "…computer readable program code which receives at a server of a plurality of servers, a common nonce that is provided to each of the plurality of servers" as claimed.

Still further, as noted in greater detail above, neither *Brezak* nor *Ganesan* teach or suggest that a server authenticates the client. Rather, as described in greater detail herein, authentication is performed by the third party authentication service 206 before granting the intermediate server (Server A) permission to delegate on behalf of the client. As such, the references, even when

combined still fail to teach or suggest "…computer readable program code which authenticates the client based on the received signed common nonce" as claimed.

Accordingly, when considering claim 32 *as a whole*, the prior art references fail to teach or suggest all of the claimed limitations. For the reasons set out above, the Board is respectfully requested to reverse the Examiner's final rejection of claim 32.

**2. Claims 2, 3, 5, 7-11, 14, 15 and 30 are Patentable Over *Brezak* in view of *Ganesan* and *Ford***

Of the Claims rejected under the second ground of rejection, Claims 2, 3, 5, 7-11, 14, 15 depend from Claim 1. Claim 30 depends from Claim 28. Thus, the appellant respectfully submits that the above dependent claims are patentable at least by virtue of their dependency upon a base claim that is believed to be allowable as set out in greater detail herein.

**2-A. Claim 2 is Patentable Over *Brezak* in view of *Ganesan* and *Ford***

Claim 2 depends from Claim 1 and further recites that obtaining a common nonce comprises generating, by an entity other than the client or the plurality of servers, a common nonce based on information obtained from each of the plurality of servers.

In making the rejection, the Examiner argues that *Ford* teaches obtaining pre-nonce contributions from a plurality of servers, citing Ford Col. 15, lines 24-31[22]. The Examiner thus concludes that it would be obvious to combine *Ford* with *Brezak* and *Ganesan* to provide a ticket granter with server nonces, combining the nonces and placing the nonces in the ticket to be signed to provide strong secret data that can be verified in the ticket[23]. As noted in greater detail above, *Brezak* and *Ganesan* fail to teach or suggest a nonce at all. While *Ford* does teach the use of a nonce in general, the combination still fails to teach or suggest all of the claimed limitations.

For example, assuming *arguendo*, that *Ford* can somehow be combined with *Brezak* and *Ganesan* to provide a common nonce based on information obtained from each of the plurality of

[22] See the final Office action mailed August 18, 2006, page 8.
[23] See the final Office action, mailed August 18, 2006, page 8.

servers to a client for signature as the Examiner suggests. The combination still fails to teach that which is claimed, because no information sent by the client is ever received at a server past the intermediary server (Server A in the *Brezak* example) as noted in greater detail above. Thus, for example, Server C in *Brezak* never receives information signed by the client, regardless of the content of the information. Thus, even if the client somehow received a common nonce as the Examiner suggests, it would not be communicated to Server C, but would rather be directed by Server A to the third party authentication service 206.

Moreover, *Ford* has nothing to do with delegation and client authentication in a multi-tiered network environment. Rather, *Ford* is directed to situations where a user needs to access private data and the client terminal that the user is logged into does not store the user's strong secret data, such as a cryptographic key used to decrypt the user's private data nor does the user have the strong secret data memorized. Accordingly, *Ford* discloses an approach that permits a client terminal to regenerate a user's strong secret data from weak secret data with the assistance of one or more servers such that resistance to attacks on the servers are maintained.

In other words, *Ford* teaches a way to split up strong secret information across multiple servers so that the strong information can be remotely stored from the user in a way that is resistant to attacks to discover the strong secret information. Essentially in *Ford*, an "initializing system and a recovery system are provided[24]. In operation, it is assumed that a user 110 has strong secret data that is desired to be used from a recovery client of the recovery system. However, the recovery client 220 does not have access to the strong secret data, so the strong secret data must be regenerated. The user enters weak secret information into the recovery client, e.g., a password, which is used to regenerate the strong secret data, e.g., a cryptographic key. The strong secret data is used to discover the user's private data, which is stored in a suitably accessible storage location.

To generate the strong secret data, a plurality of secret holding servers are used to each store generated server secret data b(i)[25]. The generating client 120 can compute the user's strong

[24] See for example, *Ford*, Col. 7, lines 4-42.
[25] See for example, *Ford*, Col. 9, lines 7-9.

secret data K as a function of the user's weak secret information and the server secret data b(i). In addition, verifier data v(i) is utilized to enable verification servers (which may be the same as the secret holding servers) to determine whether or not the recovery client 220 was able to successfully identify the user's strong secret data[26].

Notably, in one disclosed embodiment of *Ford*, the recovery client re-computes the verifier data as a one-way function of the strong secret data, and then computes the proof data as a one way function of one or more data items that include the recomputed verifier data and a non-secret nonce, e.g., a timestamp or data sent in a previous message. Each verification server computes its own proof data by applying the same one-way function to its copy of the verifier data and the nonce. If the proof data computed by the recovery client 220 matches the proof data computed by the verification server 130, the strong secret data is verified[27].

Assume that each secret holding server receives server request data M. After verifying that the server request data satisfies predefined thresholds, etc., each holding server computes a response based upon the request data M and its secret holding server secret data b(i) as c(i)=Mb(i)[28]. The secret holding server may also generate a nonce, designated as index n(i) for an instantiation of the recovery process and transmit the nonce to the recovery client 220. Each secret holding server transmits a single message that is based upon the server response data c(i) and the nonce n(i)[29].

The recovery client computes a secret component K(i) from the response c(i) received from each secret holding server message[30]. The various secret components K(i) are then combined to result in the strong secret data K[31]. The combination of the various secret components K(i) is also shown in Boxes 642, 644 and 646 of Fig. 6 of *Ford*, which is reproduced below. Where the strong secret data K comprises a cryptographic key, the cryptographic key can

---

[26] See for example, *Ford*, Col. 9, lines 43-55.
[27] See for example, *Ford*, Col. 11, lines 48-61.
[28] See for example, *Ford*, Col. 15, lines 8-23.
[29] See for example, *Ford*, Col. 15, lines 23-38.
[30] See for example, *Ford*, Col. 15, lines 43-44.
[31] See for example, *Ford*, Col. 15, lines 50-51.

then be used to recover the user's private key[32]. If the user's public key is used as the verifier data v(i), the recovery client 220 can digitally sign the various nonces n(i) using the user's private key recovered using the generated strong secret data, e.g., the private key. Each verification server verifies the digital signature using the user's public key and then verifies that the correct nonce n(i) is included in the message received thereby. Alternatively, the proof data can be computed according to the expression g(v(i), n(i)) where g is a one-way function such as a cryptographic hash function[33].

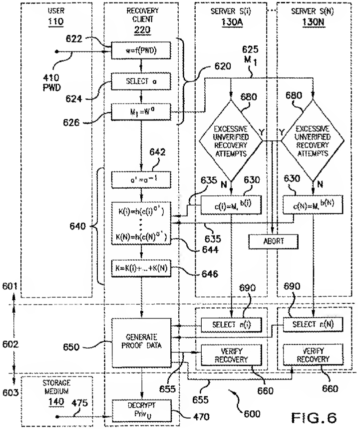



FIG.6

*Ford* does not teach or suggest providing a common nonce (signed by a client) to the plurality of servers as a signature for transactions so as to authenticate the client to the plurality of servers as claimed. Rather, each nonce n(i) is used to verify that an associated holding server

[32] See for example, *Ford*, Col 15, lines 42-55.
[33] See for example, *Ford*, Col. 15, line 56-Col. 16, line 5.

response c(i) contributed to the assembly of strong secret holding data which is used to decrypt a private key, which in turn corresponds to a public key of each verification server.

Moreover, *Ford* teaches that before an attempt to reconstruct the users strong secret data is initiated, i.e., before the nonces n(i) are even generated the user is *already authenticated* to each of the secret holding servers S(i)[34]. As such, there would be no need to provide a signed common nonce to the plurality of servers as a signature for transactions so as to authenticate the client to the plurality of servers as the ***user is already authenticated***.

Accordingly, when considering claim 2 *as a whole*, the prior art references fail to teach or suggest all of the claimed limitations. For the reasons set out above, the Board is respectfully requested to reverse the Examiner's final rejection of claims 2, 3, 5, 7-11, 14 and 15.

### 3. Claims 4, 6, 12, 13 and 20 are Patentable Over *Brezak* in view of *Ganesan*, *Ford* and *Schneier*

Each of the above rejected claims depends from Claim 1. Thus, the appellant respectfully submits that the above dependent claims are patentable at least by virtue of their dependency upon a base claim that is believed to be allowable as set out in greater detail herein.

Moreover, the Examiner only cites *Schneier* for the teaching of hashing a message when digitally signing the message and encrypting the hash with a private key as the signature rather than encrypting the whole message. The Examiner further cites *Schneier* for teaching that a verifier hashes the message, decrypts the signed hash and verifies that the two hashes are the same[35].

Regardless of how a message is hashed or encrypted, *Brezak* in view of *Ganesan*, *Ford* and *Schneier* fail to teach every element of the claimed invention. For example, *Schneier*, neither alone or in combination with *Brezak* in view of *Ganesan*, *Ford*, teach or suggest obtaining a common nonce associated with each of the plurality of servers from an entity other

[34] See *Ford*, Fig. 3, box 310; Col 4.lines 35-37; Col. 8, lines 50-53; Col. 13, lines 16-19
[35] See final Office action mailed August 18, 2006, pages 9-10.

than the client or the plurality of servers; providing the common nonce to the client; receiving the common nonce signed by the client at the middle-tier server and providing the signed common nonce to the plurality of servers as a signature for transactions so as to authenticate the client to the plurality of servers.

Accordingly, when considering each of claims 4, 6, 12, 13 and 20 *as a whole*, the prior art references fail to teach or suggest all of the claimed limitations. For the reasons set out above, the Board is respectfully requested to reverse the Examiner's final rejection of claims 4, 6, 12, 13 and 20.

### 4. Claims 16-19 and 21-22 are Patentable Over *Brezak* in view of *Ganesan*, *Ford* and *Menezes*

Each of the above rejected claims depends from Claim 1. Thus, the appellant respectfully submits that the above dependent claims are patentable at least by virtue of their dependency upon a base claim that is believed to be allowable as set out in greater detail herein.

Moreover, the Examiner only cites *Menez* for teaching that nonce challenges can be random numbers and that when using nonce challenges, the challenger should apply a timeout period to the nonce and not authenticate the client if the response is received after the timeout period has expired, and to teach that when using a certificate for authentication, the certificate expiration data can be checked, certificate revocations can be checked and if the certificate passes the checks, then the public key can be determined valid[36].

Regardless of how a certificate or nonce is encoded, *Brezak* in view of *Ganesan*, *Ford* and *Menezes* fail to teach every element of the claimed invention. For example, *Menezes*, neither alone or in combination with *Brezak* in view of *Ganesan*, *Ford*, teach or suggest obtaining a common nonce associated with each of the plurality of servers from an entity other than the client or the plurality of servers; providing the common nonce to the client; receiving the common nonce signed by the client at the middle-tier server and providing the signed common

[36] See final Office action mailed August 18, 2006, Pages 11-12.

nonce to the plurality of servers as a signature for transactions so as to authenticate the client to the plurality of servers.

Accordingly, when considering each of claims 16-19 and 21-22 *as a whole*, the prior art references fail to teach or suggest all of the claimed limitations. For the reasons set out above, the Board is respectfully requested to reverse the Examiner's final rejection of claims 16-19 and 21-22.

*Conclusion*

For all of the above reasons, the appellant respectfully submits that claims 1-32 define patentably over the applied prior art. Accordingly, it is respectfully requested that the Board reverse the Examiner's final rejection of claims 1-32.

Respectfully submitted,

Stevens & Showalter, L.L.P.
By /Thomas E. Lees/
Thomas E. Lees Reg. No. 46,867

7019 Corporate Way
Dayton, Ohio 45459-4238
Phone 937-438-6848
Fax 937-438-2124

## *Appendix A – Claims Appendix*

1. (Previously Presented) A method for a middle-tier server to impersonate a client to a plurality of servers, the method comprising:

obtaining a common nonce associated with each of the plurality of servers from an entity other than the client or the plurality of servers;

providing the common nonce to the client;

receiving the common nonce signed by the client at the middle-tier server; and

providing the signed common nonce to the plurality of servers as a signature for transactions so as to authenticate the client to the plurality of servers.

2. (Previously presented) The method of Claim 1, wherein the step of obtaining a common nonce comprises the step of generating, by an entity other than the client or the plurality of servers a common nonce based on information obtained from each of the plurality of servers.

3. (Original) The method of Claim 2, wherein the step of generating a common nonce comprises the steps of:

obtaining pre-nonce contributions from the plurality of servers;

combining the pre-nonce contributions to provide a single pre-nonce token; and

providing the common nonce based on the pre-nonce token.

4. (Original) The method of Claim 3, wherein the step of providing the common nonce comprises reducing the pre-nonce token to provide the common nonce.

5. (Original) The method of Claim 3, wherein the step of combining the pre-nonce contributions to provide a single pre-nonce token comprises concatenating the pre-nonce contributions.

6. (Original) The method of Claim 4, wherein the step of reducing the pre-nonce token to provide the common nonce comprises the step of hashing the pre-nonce token utilizing a one-way hash function so as to provide the common nonce.

7. (Original) The method of Claim 3, wherein the step of obtaining pre-nonce contributions comprises the steps of:

requesting a pre-nonce contribution from each of the plurality of servers; and receiving the pre-nonce contributions from the plurality of servers.

8. (Original) The method of Claim 7, wherein requesting a pre-nonce contribution comprises sending authenticated requests to the plurality of servers.

9. (Original) The method of Claim 8, further comprising the step of encrypting the authenticated requests sent to the plurality of servers.

10. (Original) The method of Claim 8, wherein the authenticated requests include at least one of an identification of a source of the request, a time stamp and a random number.

11. (Original) The method of Claim 3, wherein the pre-nonce contributions include at least one of an identification of a server of the plurality of servers and a random number.

12. (Original) The method of Claim 3, wherein the pre-nonce contributions are signed with a signature corresponding to a server from which the pre-nonce contribution was obtained, the method further comprising incorporating the signatures in the pre-nonce token.

13. (Original) The method of Claim 3, wherein the pre-nonce contributions are signed with a signature corresponding to a server from which the pre-nonce contribution was obtained, the method further comprising authenticating the signatures of the pre-nonce contributions and rejecting pre-nonce contributions for which the digital signature is not authentic.

14. (Original) The method of Claim 3, further comprising the steps of:

receiving a transaction identification from a trusted server of the plurality of servers; and associating the transaction identification with the common nonce.

15. (Original) The method of Claim 14, further comprising the step of tracking use of the common nonce based on the transaction identification.

16. (Original) The method of Claim 3, further comprising the steps of:

associating an expiration lime with a pre-nonce contribution; and

determining if the pre-nonce contribution has expired based on its associated expiration time.

17. (Original) The method of Claim 16, further comprising the steps of:

receiving the common nonce at a server of the plurality of servers;

determining a pre-nonce contribution associated with the received common nonce; and

accepting the received common nonce if the associated pre-nonce contribution has not expired.

18. (Original) The method of Claim 3, wherein at least one of the plurality of servers carries out the steps of:

receiving a client certificate;

determining if the client certificate is trusted; and

indicating that the client is not authenticated if the client certificate is not trusted.

19. (Original) The method of Claim 3, wherein at least one of the plurality of servers carries out the steps of:

receiving the signed common nonce and a client certificate;

determining if the signature of the signed common nonce corresponds to a signature of the client certificate; and

indicating that the client is not authenticated if the signature of the signed common nonce does not correspond to the signature of the client certificate.

20. (Original) The method of Claim 6, wherein at least one of the plurality of servers carries out the steps of:

receiving the signed common nonce, the common nonce and the pre-nonce token;

hashing the received pre-nonce token;

comparing the hashed pre-nonce token to the common nonce;

indicating that the client is not authenticated if the hashed pre-nonce token is different from the common nonce.

21. (Original) The method of Claim 11, wherein at least one of the plurality of servers carries out the steps of:

receiving the pre-nonce token;

determining if the pre-nonce token includes a random number associated with the at least one of the plurality of servers; and

indicating that the client is not authenticated if the pre-nonce token does not include the random number associated with the at least one of the plurality of servers.

22. (Original) The method of Claim 21, wherein at least one of the plurality of servers carries out the steps of:

associating an expiration with the random number associated with the at least one of the plurality of servers; and

indicating that the client is not authenticated if the pre-nonce token does not include a random number associated with the at least one of the plurality of servers which has not expired.

23. (Original) The method of Claim 1, wherein the step of obtaining a common nonce comprises the steps of:

obtaining the common nonce from a party trusted by the middle-tier server and the plurality of servers, the common nonce being signed by the trusted party; and

verifying the signature of the common nonce is the signature of the trusted party.

24. (Original) The method of Claim 23, wherein at least one of the plurality of servers canies out the steps of:

receiving a client certificate;

determining if the client certificate is trusted; and

indicating that the client is not authenticated if the client certificate is not trusted.

25. (Original) The method of Claim 23, wherein at least one of the plurality of servers carries out the steps of:

receiving the signed common nonce and a client certificate;

determining if the signature of the signed common nonce corresponds to a signature of the client certificate; and

indicating that the client is not authenticated if the signature of the signed common nonce does not correspond to the signature of the client certificate.

26. (Previously Presented) A system for a middle-tier server to impersonate a client to a plurality of servers, comprising:

means for obtaining a common nonce associated with each of the plurality of servers from an entity other than the client or the plurality of servers;

means for providing the common nonce to the client;

means for receiving the common nonce signed by the client at the middle-tier server; and

means for providing the signed common nonce to the plurality of servers as a signature for transactions so as to authenticate the client to the plurality of servers.

27. (Previously Presented) A computer program product for a middle-tier server to impersonate a client to a plurality of servers, comprising:

a computer readable media having computer readable program code embodied therein, the computer readable program code comprising:

computer readable program code that obtains a common nonce associated with each of the plurality of servers from an entity other than the client or the plurality of servers;

computer readable program code that provides the common nonce to the client;

computer readable program code that receives the common nonce signed by the client at the middle-tier server; and

computer readable program code that provides the signed common nonce to the plurality of servers as a signature for transactions so as to authenticate the client to the plurality of servers.

28. (Previously Presented) A method of authenticating a client, comprising:

receiving at a server of a plurality of servers, a common nonce that is provided to each of the plurality of servers from an entity other than the client or the plurality of servers, the common nonce being associated with each of the plurality of servers and signed by the client; and

authenticating the client based on the received signed common nonce.

29. (Original) The method of Claim 28, wherein the common nonce is provided by a trusted third party.

30. (Previously Presented) The method of Claim 28, wherein the common notice is generated by an entity other than the client or the plurality of servers based on information provided by each of the plurality of servers.

31. (Previously Presented) A system for authenticating a client, comprising:

means for receiving at a server of a plurality of servers, a common nonce that is provided to each of the plurality of servers from an entity other than the client or the plurality of servers, the common nonce being associated with each of the plurality of servers and signed by the client; and

means for authenticating the client based on the received signed common nonce.

32. (Previously Presented) A computer program product for authenticating a client, comprising:

a computer readable media having computer readable program code embodied therein, the computer readable program code comprising:

computer readable program code which receives at a server of a plurality of servers, a common nonce that is provided to each of the plurality of servers from an entity other than the client or the plurality of servers, the common nonce being associated with each of the plurality of servers and signed by the client; and

computer readable program code which authenticates the client based on the received signed common nonce.

## *Appendix B – Evidence Appendix*

There is no information for this appendix

## *Appendix C – Related Proceedings Appendix*

There is no information for this appendix